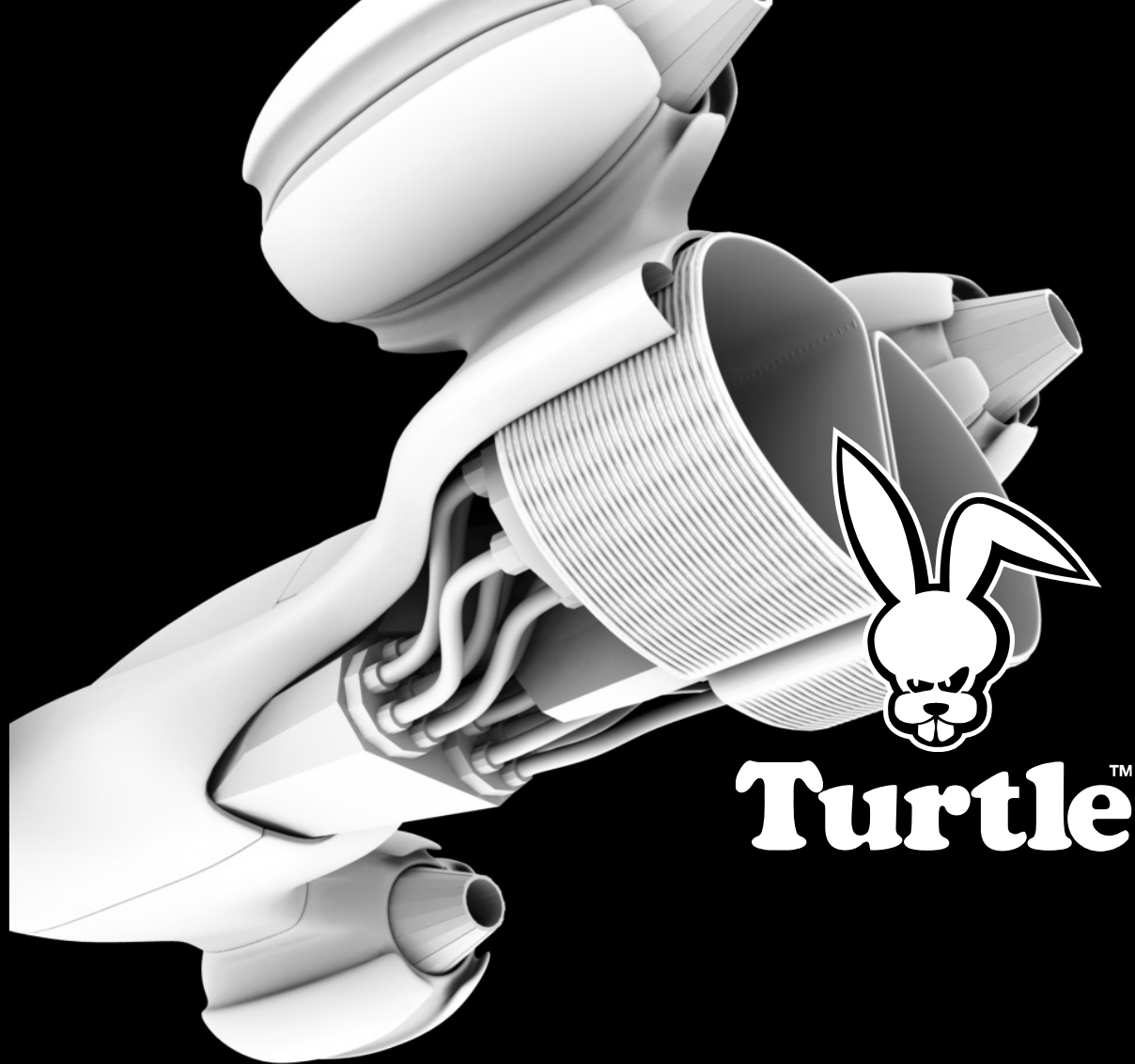


Ambient and Reflection Occlusion



In this tutorial we are going to check out just how fast Ambient Occlusion in Turtle really is.



ILLUMINATE LABS™

LIGHTING THE NEXT GENERATION OF GAMES

Ambient Occlusion is a trick invented by film makers to get an indirect illumination effect in a controllable way. For every surface point, you sample the surrounding geometry in every direction, and estimate how much incoming light is blocked from reaching the surface point. You can either use the data directly in your shader networks, or bake it down into a map to avoid rerenders or for game content.

Ambient Occlusion is a great tool for adding subtle shading to your images. It's difficult to tune your final gather renders to be really flicker-free, so approximating a similar effect with Ambient Occlusion maps is a great way to avoid flicker.



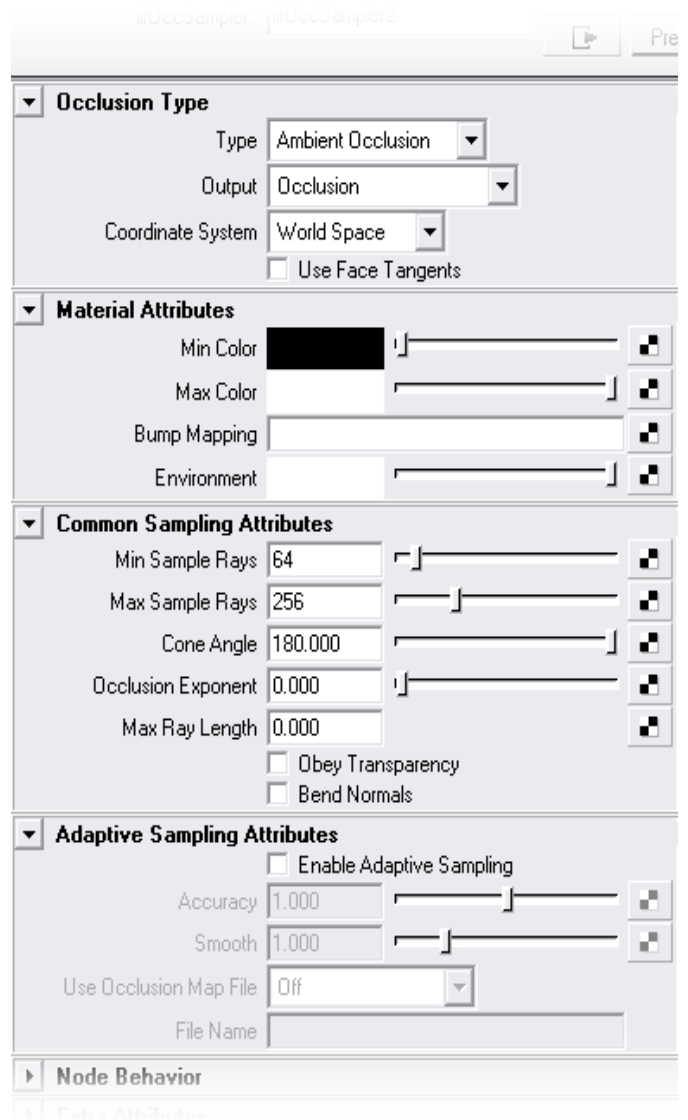
Go ahead and open the provided scene [craft.mb](#) in Maya and render out a frame. This scene is setup with a simple reflective Blinn shader, illuminated by some basic Maya lighting.



0:23 min

Ambient Occlusion

Select the craft group in the Outliner and assign the existing `s_ambOccl` shader to the surfaces. The `Ir Occ Sampler` shader controls both **Ambient Occlusion** and **Reflection Occlusion**, but the defaults should be OK for now. You don't normally connect the shader directly as the main material, but it's a great way to get to know the attributes that control the shader. Fire up a render again.



We get a really nice clay look in the render, but the render time increased a lot. For every sample in the image, Turtle is currently set to sample between 64 and 256 times to estimate the Ambient Occlusion. This is quite a lot of sampling, but luckily, Turtle has a really smart mode where slowly varying areas can be less densely sampled, so that only detailed areas get a very high sampling. Go ahead and check **Enable Adaptive Sampling** and render out a new frame.



We get an almost identical image, but in a much shorter render time! **Adaptive Sampling** should really be enabled by default in the shader, so remember to always check it when you're rendering Ambient Occlusion. There's really no reason not to.

There are quite a lot of attributes to control the occlusion shader, but you really only need to remember and know a few of them for your everyday work. Attributes like [Output](#) and [Coordinate System](#) rarely need changing, since they let you write out different components or choose another vector basis for the output, so the defaults are fine for basic ambient occlusion. [Use Face Tangents](#) controls whether the coordinate space is interpolated from vertices, or calculated directly at the sample point. This too will only matter if you specifically need Face Tangents for some reason in your project, otherwise, don't worry about it.

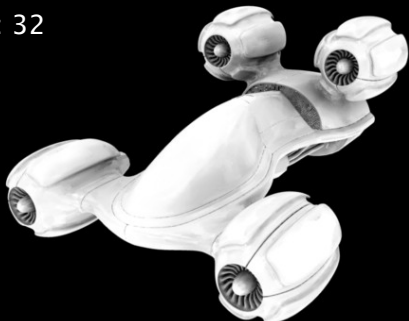
Material Attributes

The trick with Ambient Occlusion is that it's completely independent of materials and lighting in the scene, so you can reuse any baked Ambient Occlusion as long as the geometry remains unchanged. You still have some options with which to control the output, however. You can for instance change the color gradient of the Ambient Occlusion through the [Min Color](#) and [Max Color](#) attributes. The Ambient Occlusion sampler does not normally honor any material attributes, but you do have the option to connect a bump map through the [Bump Mapping](#) attribute. The [Environment](#) channel lets you connect an environment shader for image based ambient lighting. You will also frequently use it together with Reflection Occlusion.

Sample Rays

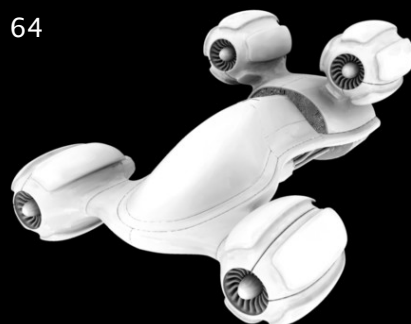
For every Ambient Occlusion sample point in the image, the Sample Rays attributes control how many rays are used to probe the surrounding geometry. Turtle will automatically sample the optimal amount of rays between the [Min Sample Rays](#) and [Max Sample Rays](#) values. The default values of [Min Sample Rays](#): 64 and [Max Sample Rays](#): 128 are in most cases a good start. If you go lower, you will notice an increase in noise. You might notice that to retain finer details, you will have to raise the sampling for final renderings. Even [Min Sample Rays](#): 512 and [Max Sample Rays](#): 1024 is pretty fast with the Adaptive Sampling mode.

Min: 16
Max: 32



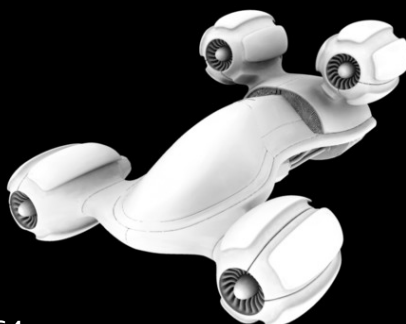
0:32 min

Min: 32
Max: 64




0:34 min

Min: 64
Max: 128

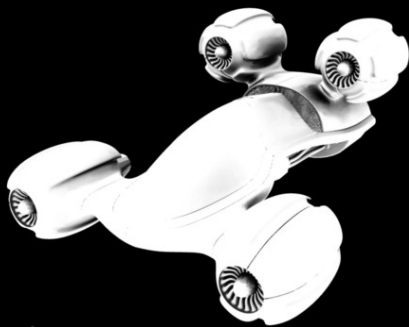


0:38 min

Cone Angle

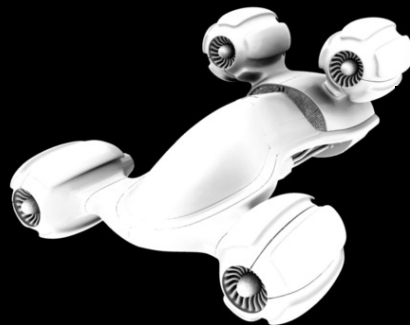
 The **Cone Angle** limits the distribution of the Ambient Occlusion rays. The default value of 180.0 sends rays in the hemisphere above the surface point. You can tweak the value to get a more limited ambient occlusion effect, so try out a few renders and see what look you like.

Cone Angle: 120.0



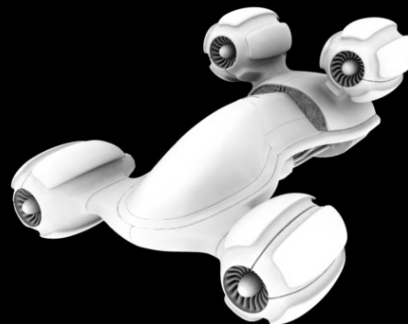
0:32 min

Cone Angle: 150.0




0:36 min

Cone Angle: 180.0



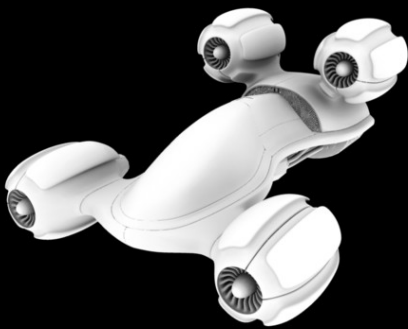
0:38 min

 There are some more attributes here, but you'll rarely touch them. **Occlusion Exponent** can be used to boost the Ambient Occlusion gradient, so that the transition from **Min Color** to **Max Color** is sharper. **Max Distance** is set by default to 0.0, which means that there is no restriction on how far Turtle will sample Ambient Occlusion in the scene. By actually setting it to a small value, you will reduce the render time, because Turtle will be able to skip out of the sampling earlier. Check **Obey Transparency** if there are transparent surfaces like glass in the scene, otherwise surfaces might be improperly shaded. **Bend Normals** is important if you are outputting vectors. By enabling this function, the vectors will change from the surface normals to the dominant incoming light direction, which Turtle estimates through the occlusion samples.

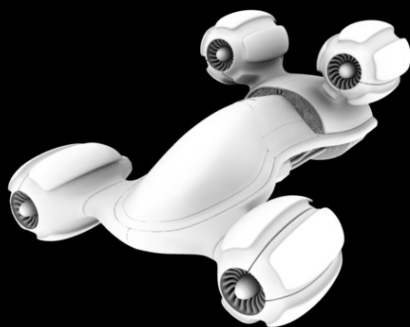
Adaptive Sampling

The **Adaptive Sampling** mode works exactly as a Final Gather rendering, it runs a rough pre-pass and samples Ambient Occlusion adaptively. During the final rendering, the Ambient Occlusion values are interpolated from the prepass results. You control the detail of the pre-pass with the **Accuracy** attribute, and the smoothness of the interpolation with the **Smooth** attribute. The **Accuracy** and **Smooth** values are still dependent on each sample point being sufficiently sampled, so a high **Accuracy** or **Smooth** value can not counter low **Sample Rays** settings.

One of the coolest things about using **Adaptive Sampling** is that you can automatically reuse earlier Ambient Occlusion data. Simply change the **Use Occlusion Map File** to **Reuse and append** and set a filename to store the data in. This way Turtle will reuse previous calculations in your next render and append extra occlusion data in case it's needed. You will notice the most savings in render time when you are using quite high Ambient Occlusion settings. Remember however, that you can only reuse calculations for static objects!



- Min Sample Rays: 512
- Max Sample Rays: 1024
- Use Occlusion Map File: off
- Render time: 1:04 min




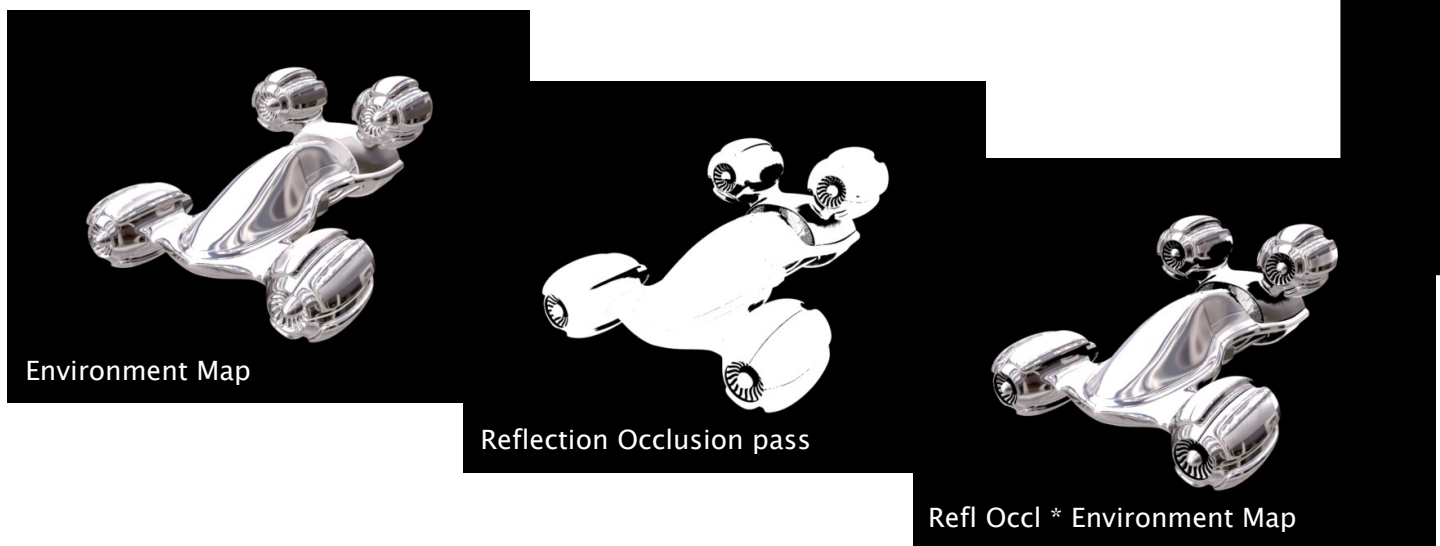
- Min Sample Rays: 512
- Max Sample Rays: 1024
- Use Occlusion Map File: Reuse and append
- Render time: 0:33 min

Use the **Adaptive Sampling** mode whenever you can; remember to check **Reuse and append** as well, since this will save you precious render time.

Reflection Occlusion

An often used production technique is to fake reflections with environment maps, without actually ray tracing. One of the drawbacks to this method is that the reflections will show up in every part of the surface, even though some areas might be occluded by the object itself. A common workaround is to render out another pass, which masks out the occluded areas, a pass that is commonly referred to as a **Reflection Occlusion** pass. The generated images will look pretty close to Ambient Occlusion, but there are subtle differences in the sampling. Ambient Occlusion always samples in a cone around the normal of the sampled surface, whereas in Reflection Occlusion, samples are distributed in a cone around the reflected camera ray from the sampled surface.

 Assign the existing material `s_refOccl` to the craft surfaces and hit render. The **Cone Angle** is only 10.0 degrees, so the transitions are quite sharp, but for highly reflective objects, rays don't scatter much either. If you change the output to **Environment * Occlusion**, you will see directly how the Reflection Occlusion will affect an environment map. Try setting both **Min Color** and **Max Color** to white in order to remove the Reflection Occlusion map temporarily.



If you check the Sampling tab in **Render Settings**, notice that in the **Ray Tracing** roll-out, the **Reflections** setting is set to zero. This means that Turtle did not ray-trace any reflections at all, so if we were to render out the Reflection Occlusion pass as a sequence, we could reuse it later without ever ray tracing the reflections in the ship.

As you might have noticed, the Adaptive Sampling attributes in the Occlusion shader are all grayed out when using Reflection Occlusion. Adaptive Sampling does not apply well to rendering Reflection Occlusion, but luckily, you can lower the **Sample Rays** values quite a bit when the **Cone Angle** attribute is small. If you notice the `s_refOccl` shader, **Min Sample Rays** and **Max Sample Rays** are set as low as 12 and 24, and this still gives quite good results.

Blurred Reflections


You can pair a larger **Cone Angle** with a pre-blurred environment map to create a convincing glossy reflection effect. Never blur the image inside Maya, as this will cost you in render time, so always remember to pre-blur in HDR Shop or a similar tool.




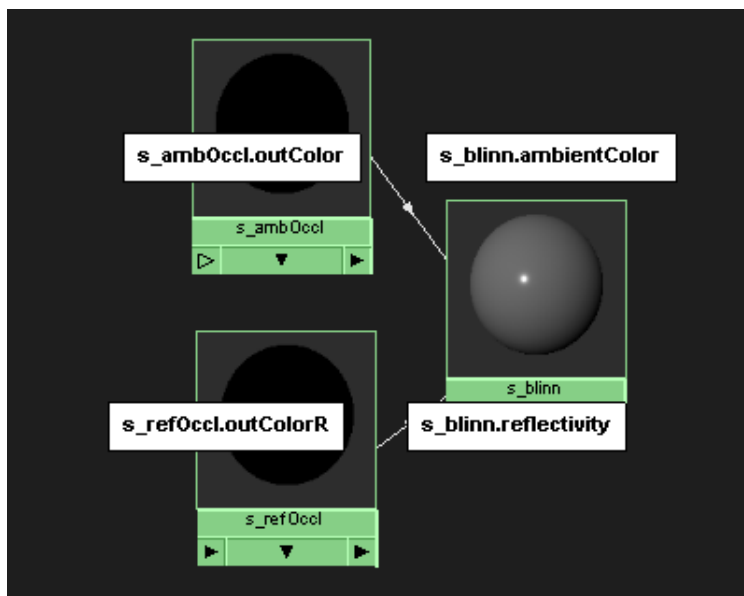
- Min Sample Rays: 64
- Max Sample Rays: 128
- Cone Angle: 120.0
- Environment Blur: 25.0


Putting it all together

You are usually best served by dividing your work into passes when you want to use Ambient and Reflection Occlusion, and then compositing all the components in a stand-alone compositing application. Just for simplicity's sake, we'll connect the equivalent inside Maya, but for production work, this is not recommended.

 Select the craft group and assign the existing shader **s_blinn**. Open the hypershader and drag **s_blinn**, **s_ambOccl** and **s_refOccl** to the workarea.

 Connect the **outColor** of the Ambient Occlusion shader to the **ambientColor** of the Blinn shader. Connect one of the **outColor** components of the Reflection Occlusion shader to the **reflectivity** of the Blinn shader. The output of both Occlusion shaders goes as high as white, which is too high for both the reflectivity and the ambient color, so change the **Max Color** of both shaders to HSV: 0.0, 0.0, 0.25.



 We're going for a sharp reflection look, so make sure that the environment map is unblurred, and the Reflection Occlusion Shader is set back to **Min Sample Rays: 12**, **Max Sample Rays: 24** and **Cone Angle: 10.0**. If you want to adjust the ambient color, it might be better to mix a color input with the Ambient Occlusion shader, but this is sort of a simplified version, where we directly set the Ambient Color through the **Min Color** and **Max Color** of the Ambient Occlusion shader. Render out a frame now and save it in the Render View. Set the **Min Color** of both Occlusion shaders to HSV: 0.0, 0.0, 0.25 so that the Occlusion shaders will output a constant color, i.e. switching off the Occlusion contribution, and render out another frame.

Blinn shader
Ambient Occlusion: off
Reflection Occlusion: off



Blinn shader
Ambient Occlusion: on
Reflection Occlusion: on



For the final render, we boost all the [Sample Rays](#) values of the Occlusion shaders and raise the resolution to 1280x960. Still no ray tracing except for the Occlusion shaders!



- **Reflection Occlusion:**
Min Ray Samples: 64
Max Ray Samples: 128
- **Ambient Occlusion:**
Adaptive Sampling
Min Ray Samples: 512
Max Ray Samples: 1024
- **Render Time: 2:18 min**

Get your eval license of Turtle now!
www.illuminatelabs.com